Aula 4 (14/10/2022) Nome: Adriel Bombonato Guidini Godinho RA: 191011631 Conversão de modelos de representação de imagens coloridas Modelos de cor, também chamados de espaço de cores e sistema de cores, têm propósito de facilitar a representação de uma cor dentro de um padrão. Um espaço de cores é uma representação de sistemas de coordenadas, em que cada ponto é uma cor específica. Variar coordenadas de diferentes espaços gera diferentes mudanças. In []: import cv2 import numpy as np from matplotlib import pyplot as plt In []: # Tipos de modelos de representação de cores presentes no Opencv flags = [i for i in dir(cv2) if i.startswith('COLOR_')] print(flags[0:20]) ['COLOR_BAYER_BG2BGR', 'COLOR_BAYER_BG2BGRA', 'COLOR_BAYER_BG2BGR_EA', 'COLOR_BAYER_BG2RGB_VNG', 'COLOR_BAYER_BG2RGB_VNG', 'COLOR_BAYER_BG2RGB_VNG', 'COLOR_BAYER_BG2RGB_VNG', 'COLOR_BAYER_BG2RGBA', 'COLOR_BAYER_BG2RGB_VNG', 'COLOR_BAYER_BGA LOR_BAYER_BGGR2BGRA', 'COLOR_BAYER_BGGR2BGR_EA', 'COLOR_BAYER_BGGR2BGR_VNG', 'COLOR_BAYER_BGGR2GBA', 'COLOR_BAYER_BGGR2RGB', ' 2BGRA'] **RGB** O sistema RBG é muito conhecido e é representado por um cubo que varia em 3 eixos: **B**lue, **G**reen e **R**ed. Red Green A imagem no Opencv é lida em BGR, portanto, devemos convertê-la para plotar em RGB. In []: #BGR: # Leitura da imagem em BGR img_BGR = cv2.imread('images\pimenta.jpg') # Conversão manual para RGB img_RGB = cv2.cvtColor(img_BGR, cv2.COLOR_BGR2RGB) # Plotar todos os resultados, sendo que o matplotlib automaticamente plota em RGB titles = ['Imagem em BGR', 'Imagem em RGB'] images = [img_RGB, img_BGR] for i in range(2): images[i] = cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB) plt.subplot(1,2,i+1),plt.imshow(images[i]) plt.title(titles[i]) plt.xticks([]),plt.yticks([]) plt.show() Imagem em BGR Imagem em RGB **HSV** O sistema HSV, ou HSB, é descrito em Hue, Saturation e Value/Brightness. Ele se destaca pela facilidade de trocar de uma cor para outra alterando apenas um valor, o que facilita o entendimento do processo. Uma cor com valor ou brilho máximo é análogo a utilizar uma luz puramente branca em um objeto colorido. In []: # HSV: # Conversão para HSV img_HSV = cv2.cvtColor(img_BGR, cv2.COLOR_BGR2HSV) # Plotar todos os resultados titles = ['Imagem em BGR', 'Imagem em RGB', 'Imagem em HSV'] images = [img_RGB, img_BGR, img_HSV] for i in range(3): images[i] = cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB) plt.subplot(1,3,i+1),plt.imshow(images[i]) plt.title(titles[i]) plt.xticks([]),plt.yticks([]) plt.show() Imagem em HSV Imagem em BGR Imagem em RGB Lab O espaço de cores CIELAB, ou Lab, expressa a cor em 3 diferentes formas. O L representa a porcentagem de Lightness, enquanto a e b cores únicas da visão humana: amarelo, vermelho, verde e azul. Lab tem a vantagem de ser mais perceptivamente linear que outros espaços de cores, o que significa que uma mudança de um valor produz uma mudança de percepção igualmente importante. In []: # Lab: # Conversão para Lab img_Lab = cv2.cvtColor(img_BGR, cv2.COLOR_BGR2LAB) # Plotar todos os resultados titles = ['Imagem em BGR', 'Imagem em RGB', 'Imagem em HSV', 'Imagem em Lab'] images = [img_RGB, img_BGR, img_HSV, img_Lab] for i in range(4): images[i] = cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB)
plt.subplot(2,2,i+1),plt.imshow(images[i]) plt.title(titles[i]) plt.xticks([]),plt.yticks([]) plt.show() Imagem em BGR Imagem em RGB Imagem em HSV lmagem em Lab Luv O CIE 1976 L, u, v*, ou Luv, foi criado com o mesmo objetivo do Lab, a uniformidade de percepção. Também opera no sistema de coordenadas CIEXYZ. A CIE recomenda a utilização do Lab para caracterização de superfícies coloridas, enquanto recomenda o Luv para displays de cores. In []: # Luv: # Conversão para Luv img_Luv = cv2.cvtColor(img_BGR, cv2.COLOR_BGR2Luv) # Plotar todos os resultados titles = ['Imagem em BGR', 'Imagem em RGB', 'Imagem em HSV', 'Imagem em Lab', 'Imagem em Luv'] images = [img_RGB, img_BGR, img_HSV, img_Lab, img_Luv] for i in range(5): images[i] = cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB) plt.subplot(3,2,i+1),plt.imshow(images[i]) plt.title(titles[i]) plt.xticks([]),plt.yticks([]) plt.show() Imagem em BGR Imagem em RGB Imagem em HSV Imagem em Lab lmagem em Luv Grayscale Imagem com apenas 1 canal, com níveis de intensidade de preto e branco. In []: # Conversão para Lab img_GRAY = cv2.cvtColor(img_BGR, cv2.COLOR_BGR2GRAY) # Plotar todos os resultados titles = ['Imagem em BGR', 'Imagem em RGB', 'Imagem em HSV', 'Imagem em Lab', 'Imagem em Luv', 'Imagem Grayscale'] images = [img_RGB, img_BGR, img_HSV, img_Lab, img_Luv, img_GRAY] for i in range(6): images[i] = cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB) plt.subplot(3,2,i+1),plt.imshow(images[i]) plt.title(titles[i]) plt.xticks([]),plt.yticks([]) plt.show()

Imagem em BGR

lmagem em HSV

lmagem em Luv

img = cv2.imread('images\\forest.jpg', 0)

Aplicando CLAHE na imagem

Plotar todos os resultados

plt.figure(figsize=(10, 7))

plt.title(titles[i])

plt.xticks([]),plt.yticks([])

cl1 = clahe.apply(img)

images = [img, cl1]

for i in range(2):

plt.show()

Imagem em RGB

Imagem em Lab

Imagem Grayscale

lmagem com equalização

Corrigir distribuição de intensidades luminosa

clahe = cv2.createCLAHE(clipLimit=30.0, tileGridSize=(8,8))

images[i] = cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB)

titles = ['Imagem original', 'Imagem com equalização']

plt.subplot(1,2,i+1),plt.imshow(images[i])

lmagem original

In []: # Usando equalização de histograma com método CLAHE para evitar saturação de intensidade